

**REVISED
2021**

ICSE CLASS 10

COMPUTER APPLICATIONS IMP NOTES

**BEST NOTES TO SCORE FULL
MARKS IN SECTION A AND B.**

**EACH TYPE OF OUTPUT
QUESTION SOLVED**

VERIFIED BY ICSEIMPNOTES

ICSE COMPUTER THEORY IN A NUTSHELL(E NNOTES)

- 1) Static – distinguishes between instance variables and class variables.
 - 2) '/': returns the quotient of two operands
 - 3) '%'- returns the remainder of two operands
 - 4) //Single line comment /*Multi line comment/
 - 5) Encapsulation: Wrapping up of data and methods in a single unit
 - 6) Types of JAVA program: Applet , application program interphase
 - 7) Object: Identifiable identity with some characteristics and behavior
 - 8) Polymorphism: It allows a function to be used for more than one time. It is the property that allows two or more classes to respond to the same message in different ways.
 - 9) Inheritance: It is a process by which a class acquires properties of another class.
 - 10) Abstraction: the act of representing essential features of an object without representing its background details.
 - 11) Method: A member function which represents some behavior of an object.
 - 12) Member variables are also called instance variables
 - 13) JDK: Java development kit. Works on DOS based platform. Error handling is difficult.
 - 14) Principles of OOP- a) Abstraction b) Encapsulation c) Polymorphism d) Inheritance
 - 15) What does values of member variables define? – state of an object
-

16) Why object called instance of class? - since an object contains all the necessary information (data members and member methods) specified inside the class

17) Scope of variable : The region within which a variable is accessible

18) Local variable: Variable declared inside a method or block

19) Class variable: variable available to the entire class

20) Visibility: refers to whether one can access a given variable from a given scope of program.

21) Visibility modifier/Access specifier: special keyword which limits the scope of visibility that is accessibility of class, variable or method

22) Visibility mode: the property which controls the visibility i.e. accessibility of data members or member methods in a class or a sub class mode.

23) Character set: A set of valid characters that a language can recognize.

24) Literals/constants: Data items that never change their value during program execution

25) Tokens: Smallest individual units in a program

26) Final/ Symbolic constant: Converts a variable into a constant.

27) Keywords: reserved words which carry a special meaning of purpose to java compiler.

28) Variable: computer memory location symbolic name

29) Unicode : two bit character set used in java

30) Import: keyword used to call a package in a program

31) Boolean < char < short < int < double

32) Operators: Symbols that trigger some operation

33)Wrapper class: Classes that correspond to each of simple types/ class that wrap the value of a primitive data type in an object.

34)Operands: Objects of operators

35)Void(): represent non returnable type of statement. Used when a function does not return any value

36)Not operator(!): used to negate the result of expression. It is a unary operator

37)New keyword: used for allocating memory to an array

38)Return: causes the control to transfer back to method call statement.

39)Constructor: A member function used to create and initialize the object with legal set of values. It is automatically invoked at the time of object creation.

40)Parameterized constructor: constructor that can take arguments.

41) Default constructor: accepts no parameter. It initializes data members by null or zero value.

42)Constructor overloading: two or more functions defined in a class.

43)This: used to refer to the current calling object.

44)Package invoked at default: java.lang;

45)Array: Collection of data of the same types that are referenced by a common name.

46)Out of bound subscript: The subscripts other than 0 to n-1 for an array having n elements, are called out of bound subscripts.

47)Exception handling: An unexpected situation may arise during execution of a program. They are known as exceptions. Such kind of situations are handled by the exception handling mechanism in Java.

48)Features of Java: i) Platform independent i.e. program developed in java can be executed anywhere on any system. ii) portable iii) object oriented

49)BlueJ: it is a java development environment . It is an IDE which includes an editor, a debugger, and a viewer.

50)Editor: use to write the programs

51)Debugger: to find the program graphically.

52)Viewer: to see parts of program differently

53)State java concept implemented through: 1) A superclass and subclass : inheritance 2) the act of communication among objects of a class: Message passing

54)Encapsulation is a way of implementing abstraction.

55)Default class: A class visible to all classes of package

56)Type conversion: Conversion of one data type into another is called type conversion.

57)Implicit type conversion: Automatic conversion of one data type into another that takes place without user's intervention.

58)Explicit type conversion: Conversion of one data type into another that takes place in accordance with the user's requirement.

59)Class as composite data type: as class is made up of a number of primitive data types

60)Why methods declared static? When a method of a class needs to be referenced without the help of their objects, they are declared as static.

61)Nested class: defining a class within another class.

62)String: Sequence of characters. All strings should be given in double quotes("")

63)Ternary operator: Operates on three operands. Also known as conditional operator. It is used as an alternative of if else statement Syntax:
variable=condition ? true value: false value;

64)Logical operator: Combines the result of two or more expressions. The result is Boolean (true or false) e.g.: &&- AND | - OR !- NOT.

65)Bitwise operator: It can be applied to the integer types long , int , short , byte. The operator act upon the individual bits of the operands. They work on integer data only. E.g.: ~- Bitwise complement &- bitwise AND ^- Bitwise XOR | - Bitwise OR.

66)Compound Statement: They are the statements which contain one or more statements inside it.E.g.: loops , methods, if() , switch.

67)Precedence of operator: It determines which operator is to be used first for evaluation.

68)New operator: it is used to dynamically create an object and assign initial value to it.

69)Class as object factory: class contain all the information which is needed to create an object, therefore , class is called as object factory.

70)Characteristics of constructors: i) Same name as class ii) they do not return any value, not even void.

71)Temporary instance of class: It means an anonymous object of same class which is short-lived.It is used when an object is required for only a very short time, we need not to reserve memory for it for a long time. It is created b an explicit call to the constructor. E.g. new time().print().

72)Autoboxing: the automatic conversion of a primitive data type to its equivalent wrapper object.

73)Unboxing: the automatic conversion of wrapper class to its corresponding primitive data type.

74)Buffering: It is temporary storage used to hold data until enough has been collected that it is worth transferring.

75)Cbrt(): return the cube root of a given numeric value. Return data type is double.

76)Ceil(): used to return the double value which is greater than or equal to passed argument and is equal to the nearest integer. E.g. : c=Math.ceil(125.9)- 126.0

77)Floor(): returns the double value which is less than or equal to passed argument and is equal to the nearest integer . e.g. c=Math.floor(125.9)- 125.0

78)Round():eg: 125.6 – 126.0

79)Variable declaration: to name a variable along with its data type

80)Variable initialization: to assign actual value to a declared variable

81)Separators/Punctuators: The symbols used to indicate where group of code are divided and arranged

82)Block: A group of null or more statements grouped together between opening and closing of brackets.

83)Dot operator(.) / member reference operator: used to access member of a class or an object.

84)Expression: A valid combination of operators and operands

85)Statement: A valid expression which terminates with a semicolon

86)Loop- A repetitive structure used in a program to operate a statement for a given number of times.

87)Nested loop- A loop inside another loop.

88)Fixed loop- When a loop is to be repeated for a definite number of times

89)Unfixed/variable loop- A loop which repeats the process till a given condition is true.

90)Null/Delay loop- A loop which does not contain any statement to repeat.

91)Infinite/Endless loop- A loop which repeats its execution endless times.

92)Jump statement- The process of unconditionally transferring control from one . point to another.

93)Break: Transfers the control out of loop body.

94)Continue: forces the next iteration to take place leaving the statements which are yet to execute.

96) `java.lang`: package used to activate language based functions.

97) Default: executes when matching case is not found.

98) Fall through: a situation which is generated due to absence of break statements . from the case statement. In this, control jumps to all the case statements until it finds a break statement within the switch statement.

99) Break- causes the control to go out of switch statement block after the execution of every case statement.

100) `System.exit(0)` : used to terminate the entire program from the place where it appears and takes the control to the last closing brace of the program

101) Java short hand operator: operator that combines an arithmetic . operator followed by assignment operator.

102) Byte code: Java compiler takes the source code program written in Java language and produces it in a special format known as byte code.

103) Assembly language: program written using symbolic name or code

104) Modularity: the process of dividing a long process into smaller modules where each module contains required set of instructions.

105) Message passing between objects: The act of assigning information contained in one object to another object of the same class is known as message . passing between objects.

106) CLASS as basis of all computation – JAVA is a pure OOP language so all required operations should be managed through the class itself . None of operations are possible without the class so class are basis of all computation.

107) CLASS as user defined data type- as class is declared and defined by the user along with member variables and member methods.

108) Need of functions – i) reuse the programming code. ii) complicity is reduced as complicated programs are divided into various functions by divide and conquer method- the process of dividing a task into various tasks and to organize them to required manner iii) hiding details

109) Function prototype: The very first line of function definition that tells us about the status of the function

110) Function signature: The list of parameter which includes name of function , name of variables, along with data type of each variable separated by commas to perform a task.

111) Function definition- The extension of function prototype that contains set of statements with in curly braces to perform a task.

112) Function calling(invoking)- The process of excuting function definition along with the argument list to run all the statements given within the curly braces to carry out a task.

Funtion prototype. —————> `int Sum (int x , int y)` —————> function signature

Public specifier	Private specifier	Protected
Allows the accessibility of data members or member methods within and outside the scope of a class.	Does not allow the accessibility of data members or member functions outside the scope of a class but can be accessed within the class	Allow accessibility of data members and member methods within the class and also by the sub class

Q) What is the return data type of

1. endsWith() : boolean
2. log(): double
3. isWhiteSpace(char ch): boolean
4. Math.random : double

Int , float – 4 bytes

long- 8 byte

double: 16 byte

Formal Parameter		Actual parameter	
Is used along with function name while defining a function to receive the value passed by its caller.		Used to pass the value to formal parameter at the time when a function is invoked.	
Call by value		Call by reference	
<ul style="list-style-type: none"> Primitive data types are passed Changes made in formal parameter does not reflect to actual parameter 		<ul style="list-style-type: none"> Non primitive data types are passed Changes made in formal parameter are automatically reflected in actual parameter. 	
Pure Function		Impure function	
<ul style="list-style-type: none"> Returns its value Does not Change state of an object Also called accessor method 		<ul style="list-style-type: none"> May or may not return its value Changes the state of object Also called mutator method 	
Linear search		Binary search	
<ul style="list-style-type: none"> Slow process Works on both sorted as well as unsorted data 		<ul style="list-style-type: none"> Fast process Works on only sorted data 	
byte	1 byte	Long	8 byte
Int	4 bytes	Float	4 byte

Short	2 byte	Double	16 bytes
Character	2 byte	Boolean	1 byte

Separator	Use
;	Used as a statement terminator
,	List separator
[]	Used as array subscript to define the size of the array
Escape sequence	The character constant with begins with a backslash(\) followed by a letter that reflects corresponding output during program execution.
\n	New line character- shifts the control to new line
\t	Behaves like a tab key
\?	A question mark

Boolean literal		Null literal
Represented by true or false		Represented by ‘/o’
Relational expression		Logical expression
A valid combination of operands and only relational operator and returns result in true or false		Expression which consists of two or more relational expressions long with logical operator and returns result in true or false
Print()		Println()
Used to print the argument within the parenthesis and <i>program control remain on the same line.</i>		Used to print the argument within the parenthesis and <i>program control transfer to the next line.</i>
next()		nextLine()
Used to input a string without spaces or word or token from the user		used to input a line of text or string from the user
If else statement <ul style="list-style-type: none">• Do not has a concise form• More obvious• More flexible i.e. it allows many statements as you want		Conditional operator <ul style="list-style-type: none">• Gives more concise and clean form.• Less obvious• Only a single value can be returned from it.
For <ul style="list-style-type: none">• Fixed iterative loop• Definite loop• Will not run if the	While <ul style="list-style-type: none">• Unfixed iterative loop• Indefinite loop• Will not run if the condition	Do-while <ul style="list-style-type: none">• Unfixed iterative loop• Indefinite loop

<ul style="list-style-type: none"> condition is false Entry controlled loop Declaration (int a) for (initialization ;test condition; upadation) { Body; } 	<ul style="list-style-type: none"> is false Entry controlled loop Declaration+initialization while(condition) { Body; Updation; } 	<ul style="list-style-type: none"> Will run atleast once even if the condition is false Exit controlled loop Declaration + initialization do { Loop body; Updation; } while(condition)
--	--	---

Source code	Object code
The set of instructions and statements written by computer programmer by using programming language to find the solution of a problem	The machine language program produced after the compilation of the source code is called object code/module/program
Runtime Error	Syntax error
The error which appears during execution of a program other than syntax error	The error which occurs during compliation of a program due to wrong use of syntax

Unary operator <ul style="list-style-type: none"> Work on one operand Operators used are ++, -- 	Binary operator <ul style="list-style-type: none"> Work on two operands. Operators used are +, -, *, / 	Ternary operator <ul style="list-style-type: none"> Works on three operands Operators used are ?,:;
If() <ul style="list-style-type: none"> Can check for >, <, ==, != It can check value of any data type Cannot handle floating point constants 	Switch() <ul style="list-style-type: none"> can check for equality. Can check only integer and character value Cannot handle floating point constants 	If-else() <ul style="list-style-type: none"> Work with all relational operator It can check any data type Can handle floating point constants

Entry controlled loop	Exit controlled loop
Conditions check at beginning of loop	Conditions check at end of loop
Searching	Sorting

<ul style="list-style-type: none"> • Process of checking whether an element is present in the array or not • Linear search, binary search 	<ul style="list-style-type: none"> • Process of arranging the data in ascending or descending order in the array • Selection sort , bubble sort
length	length()
Returns the number of elements in the array	Returns the number of characters present in the string.
isUpperCase()	toUpperCase()
<ul style="list-style-type: none"> • Used to check if a character is in upper case. • Returns Boolean 	<ul style="list-style-type: none"> • Used to convert a string/character into upper case • Returns char
Function	Constructor
<ul style="list-style-type: none"> • Can return its value • To execute it , we have to call a function • Can have any name except the class name 	<ul style="list-style-type: none"> • Cannot return its value • Runs automatically • Have same name as of class
=	==
<ul style="list-style-type: none"> • Assignment operator • Used to assign value to a variable 	<ul style="list-style-type: none"> • Equality operator • Used to compare two values for equality
Primitive data types	Composite data types
<ul style="list-style-type: none"> • Built in data types • The size of data types are fixed. 	<ul style="list-style-type: none"> • Created with help of primitive data types • The size of these data types are not fixed.
Static data members	Non-static data members
<ul style="list-style-type: none"> • Maintain a single copy for whole class • Shared by all objects of class • Also called class variables 	<ul style="list-style-type: none"> • Maintain a separate copy for each and every individual object of that class. • Also called instance variable
Static class	Non-static class
<ul style="list-style-type: none"> • Has the static modifier applied • Must access the members of is enclosing class through an object 	<ul style="list-style-type: none"> • It has access to all variables and methods of its outer class and may refer to them directly in the same way that other non-static members of the outer class do.

SECTION-B

Tips to score marks in program:-

- 1.read and understand the program in the reading time.
- 2.try to understand the logic of the program.
- 3.please use reading time to understand the program.
- 4.be calm and believe in yourself never give up.

QUESTION 1

Write a pattern program for the following:-

(a)WAP to generate a traingle till n terms.pass value of n(number of terms)as parameter

```
1
22
333
4444
55555
```

SOLUTION:-

```
import java.util.*;
public class generatetraingle
{
    public static void Traingle(int n)
    {
        for(int i=1;i<=n;i++){
            for(int j=1;j<=1;j++){
                System.out.print(""+i);
                System.out.println();
            }
        }
    }
}
```

(b)WAP to generate the given pattern using literation loop satements;

```
*
*#
*##
*###
*####
```

SOLUTION:-

Class pattern

```
{
    public static void display1()
    {
        for(int i=1;i<=5;i++)
        {
            for(intj=1;j<=1;j++)
            {
                if(j%2!=0)
                    System.out.print("++");
                else
                    System.out.print("#");
            }
        }
        System.out.println()
    }
}
```

```

    }
  }
}

```

QUESTION 2

WAP that reverses an array and stores it in the same array

```

public class reversearray
{
    public static void main(String args[])
    {
        int array[]={20,30,40,50,80,90};
        System.out.println("Original array:");
        System.out.println("[");
        int Len=array.length;
        for(int i=0;i<Len;i++)
        {
            System.out.print(" " +array[i]);
        }
        System.out.print("]");
        int tmp;
        for(int i=Len-1,j=0;i>=j;i--,j++)
        {
            tmp=array[i];
            array[j]=array[i];
            array[i]=tmp;
        }
        System.out.println("/nArray after reversing its elements:");
        System.out.print("[");
        for(int i=0;i<Len;i++)
        {
            System.out.print(" "+array[i]);
        }
        System.out.print("]");
    }
}

```

QUESTION 3

WAP a program to input a string and print out the text with the uppercase and lowercase letters reversed ,but all other character should remain same as before
example:

Input:WelCome To Home
Output:wELcoMe to hOMe

```

import java.util.*;
public class Thestring
{
    public static void main(String args[])
    {
        Scanner sc=newScanner(System.in)
        System.out.println("Enter string");
        String=sc.nextint( );
        character ch1=' ';
        for(int i=0;l<str.length();i++)

```

```

    {
        ch1=nstr.charAt(i);
        if(ch1.isupperCase(ch1))
            nstr.setCharAt(i,ch1.toLowerCase(ch1));
        else
            nstr.setCharAt(i,ch1.toUpperCase(ch1));
    }
    System.out.println(nstr);
}
}

```

QUESTION 4

WAP a program that performs following operation on a 2D array(2*3)

- i. Replace every even element with its square in a new array
- ii. Replace every odd element with its cube in a array

```

public class process2Darray
{
    public static void main(String args[])
    {
        int array[][]={{1,3,5},{2,4,6}}; //given 2*3 array//
        int newArr[][]=new int[2][3];
        System.out.println("your original 2D array:");
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<3;j++)
            {
                System.out.print(array[i][j]+"");
            }
            System.out.println();
        }
        for(int i=0;i<=1;i++)
        {
            for(int j=0;j<=2;j++)
            {
                if(array[i][j]%2==0)
                    newArr[i][j]=array[i][j]*array[i][j];
                else
                    newArr[i][j]=array[i][j]*array[i][j]*array[i][j];
            }
        }
        System.out.println("New 2D array after changes:");
        for(int i=0;i<=0;i<2;i++)
        {
            for(int j=0;j<3;j++)
            {

```

```

        System.out.print(newArr[i][j]+"");
    }
    System.out.println( );
}
}
}

```

QUESTION 5

WAP a program that calculate areas of geometric shapes; Traingle, Rectangle, Parallelogram

Traingle Area= $1/2*b*h$

Rectangle Area= $w*h$

Parallelogram Area= $b*h$

```

import java.util.*;
public class area
{
    public void calcArea()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Area Calculation");
        System.out.println("1.Traingle");
        System.out.println("2.Rectangle");
        System.out.println("3.parallelogram");
        System.out.println("4.Exit");
        System.out.println("Enter your choice(1...4):");
        int ch=sc.nextInt();
        double area,base,w,h;
        switch(ch) {
            Case 1:
                System.out.println("Enter base of traingle:" )
                base=sc.nextDouble();
                System.out.println("Enter height of traingle:")
                ht=sc.nextDouble();
                area=0.5*base*ht;
                System.out.println("Area of traingle is:"+area)
                break;
            Case 2:
                System.out.println("Enter width of rectangle :");
                wid=sc.nextDouble();
                System.out.print("Enter height of rectangle:");

```



```

ht=sc.nextDouble();
        area=wid*ht;
        System.out.println("Area of rectangle is:"+area);
        break;
    Case 3:
        System.out.print("Enter base of parallelogram:");
        base=sc.nextDouble();
        System.out.println("Enter height of parallelogram:");
        ht=inp.nextDouble();
        area=base*ht;
        System.out.println("Area of parallelogram is:"+area);
        break;
    Case 4:
        System.out.println("Exiting....");
        break;
    }
}
}

```

QUESTION 6

Define a class autometer have the following description

Data members/instance variables

int taxino- to store taxi number

string name - to store passenger name

int km-to store number of kilometer travelled

Member functions:

taximeter()-constructor to initialize taxino to 0,name to " " and b to 0.

input()-to store taxino,name,km

calculate()-to calculate bill for a customer according to give condition

Kilometer travelled

rate/km

1.equals to 1km

Rs25

2.more than 1km

Rs10

3.more than 10km

Rs15

display()-to display the details in the following format

Taxino	Name	Kilometers Travelled	Bill amount
--------	------	----------------------	-------------

—

—

—

—

create an object in the main method and call all above mentioned

QUESTION 7

WAP to find sum of the following series depending on user choosing 1 or 2

1. $S = 1/4 + 1/8 + 1/12 + \dots$ upto n terms

2. $S = 1/1! + 2/2! + 3/3! + \dots$ upto n terms

where ! stands for factorial of the number and factorial value of a number is the product of all integers from 1 to that number e.g. $5! = 1 * 2 * 3 * 4 * 5$
(use switch statement)

QUESTION 8

WAP to calculate sum of all prime numbers between the range of 1 and 100

QUESTION 9

WAP to print first 15 terms of fibonacci series 0 1 1 2 3 5 8 13.....

QUESTION 10

WAP to calculate rowsum and coloum sum of 2D array



ICSE IMP NOTES

ICSE COMPUTER OUTPUT QUESTIONS

	Questions	Output
1.	System.out.print("Best") System.out.println("Of luck")	Best Of luck
2.	What are the values stored in variable x and y below? double a = -6.35; double b= 14.74; double x= Math.abs(Math.ceil(a)); double y= Math rint(Math.max(a,b));	x=6.0 y=15.00
3.	Give the output of the following program segment double x=2.9,y=2.5; System.out.println(Math.min(Math.floor(x),y)); System.out.println(Math.min(Math.ceil(x),y));	2.0 3.0
4.	Write a java statement to input the following from the user using the keyboard i) Character ii) string	i) char ch; Scanner sc= new Scanner (System.in) ch= sc.nextChar() ii) String S; S=sc.nextLine()
5.	What is the output of the following program segments i) System.out.println("four:"+4+2); System.out.println("four: "+ (2+2)); ii) String S1= "Hi"; String S2= "Hi"; String S3= "there"; String S4= "HI"; System.out.println(S1+ "equals" + S2 + ""+ S1.equals(S2)); System.out.println(S1+ "equals" + S3 + ""+ S1.equals(S3)); System.out.println(S1+ "equals" + S2 + ""+ S1.equals(S4)); System.out.println(S1+ "equalsIgnoreCase" + S4 + ""+ S1.equalsIgnoreCase(S4));	four:4 2 four:4 Hi equals Hi true Hi equals there false Hi equals HI false Hi equalsIgnoreCase HI true
6.	Find the output of the following program segment i) val=200 System.out.println(val); ii) val=1600 int val, sum, n=550; sum= (n+val)>1750?400:200; System.out.println(sum);	i) 200 ii) 400
7.	Give the output of the following a) System.out.println(Math.cbrt(125)); b) System.out.println(Math.cbrt(0.027)); c) double n=3.56; System.out.println(Math.ceil(n)); d) double n=7.86; System.out.println(Math.floor(n)); e) double n=9.4; System.out.println(Math.round(n)); f) double m=9.99; System.out.println(Math.floor(m)); g) double n=5.01 System.out.println(Math.ceil(n)); h) double j=8.5 ; System.out.println(Math.round(j)); i) System.out.println(Math.ceil(7.9));	a) 5.0 b) 0.3 c) 4.0 d) 7.0 e) 9.0 f) 9.0 g) 6.0 h) 9.0 i) 8.0

8.	What will be the output of the following code? <pre> class Test { public static int v=10; public static void m1() { int v=25; System.out.println(v); } public static void m2() { System.out.println(v); } public static void main(String args[]) { m1(); m2(); } </pre>	25 10 Here local variable v hides the class variable
9.	What is the final value of c after execution of the loop <pre> int c=0; for (int t=1;t<=5;t++) for (int j=1;j<=5;j+=2) ++c; </pre>	c= 15
10.	System.out.println("Incredible" + \n + "world");	Incredible world
11.	State the data type and values of a and b after the following segment is executed <pre> String s1= "Computer", s2= "Applications"; a= (s1.compareTo(s2)); b= (s1.equals(s2)); </pre>	int type, a=2; boolean type, b=false
12.	<pre> public static void main(String [] args) (imp) { int a=5; a++; System.out.println(a); A--(a--) - (--a) System.out.println(a); } </pre>	6 4
13.	<pre> int k=5,j=9; k+=k++ - ++j +k; System.out.println("k="+k); System.out.println("j="+j); </pre>	k=6 j=10
14.	<pre> double b= -15.6 ; double a= Math rint(Math.abs(b)); System.out.println("a="+a) </pre>	a=16.0
15.	<pre> int a=0, b=30, c=40; a= b+ c++ +b System.out.println("a="+a); </pre>	a=100;
16.	int a=0, b=30 , c=40;	98

	<pre>a = --b + c++ + b System.out.println(a);</pre>	
17.	Find the output of the following , when: <pre>val=500 val=1600 int val,sum,n=550; sum= n+ (val>1750?400:200; System.out.println(sum);</pre>	i) 750 ii) 750
18.	What will be the output if x=5 initially, i) 5* x++ ii) 5*++x	i) 25 ii) 30
19.	<pre>char c= 'A'; short m=26; int n = c+m; System.out.println(n);</pre>	91
20.	<pre>int res= "A". What is the value of A?</pre>	res=65
21.	<pre>char ch= 'F'; int m=ch; m=m+5; System.out.println(m+ "" + ch);</pre>	75 F
22.	<pre>String A= "26" , B= "100" String D= A + B + "200"; int x= Integer.parseInt(A); int y= Integer.parseInt(B); int d= x+y; System.out.println("Result 1 =" +D); System.out.println("Result 2=" +d);</pre>	Result 1 = 26100200 Result 2= 126
23.	<pre>class test { public void check() { int fine =0; boolean paid= false; fine= paid?0:200; System.out.println(fine) } }</pre>	200
24.	<pre>int a=10 , b=15 , c=20; i) System.out.println("Result" + (a>=10)); ii) System.out.println("Result" + !(a>=10)); iii) System.out.println("Result" + (a+5>c); iv) System.out.println("Result" + ((a+c)==(b+15)));</pre>	i) Result true ii) Result false iii) Result false iv) Result true
25.	<pre>int a=10, b=20 , c=9; a= --c + b++ +a; System.out.println(a);</pre>	38
26.	In m=5 and n=2 output the values of m and n after execution in (i) and (ii)	i) 3 ii) 7.5 if m and n are double, 7 if m and n is int.

	i) $m = n$ ii) $n = m + m/n$	
27.	<pre> class Main() { Main() { calculate() System.out.println("constructor"); } void calculate() { show() System.out.println("calculating"); } void show() { System.out.println(" I am displaying"); } public static void main() { Main ob= new Main(); } } </pre>	I am displaying calculating constructor.
28.	What is the return type of i) compareTo() ii) equals()	i) int ii) Boolean
29.	<pre> String s= "Examination"; int n=s.length(); System.out.println(s.startsWith(s.substring(5,n))); System.out.println(s.charAt(2)==s.charAt(6)); </pre>	false true
30.	<pre> String s= "malayalam"; System.out.println(s.substring(1).length-8); System.out.println(s.substring(1).length); </pre>	0 8
31.	<pre> String n= "Computer Knowledge."; String m= "Computer Applications"; System.out.println(n.substring(0,8).concat(m.substring(9))); System.out.println(n.endsWith("e")); </pre>	Computer Applications true
32.	<pre> public boolean f(int a , int b) { boolean e= false; while (a>1 && b>1) { if(a>b) a=a-b; else b=b-a; } if (a==1 b==1) </pre>	i) true ii) false

	<pre>e=true; return e; } i) f(28,39) ii) f(27,39)</pre>	
33.	<pre>System.out.println("Character.isUpperCase('R')); System.out.println("Character.toUpperCase('j'));"</pre>	i) true ii) J
34.	<pre>System.out.println(Math.max(-17,-19)); System.out.println(Math.ceil(7.8));"</pre>	-17 8
35.	<pre>How to swap two numbers using any third variables: int t; t=a; a=b; b=t; System.out.println("a="+a); System.out.println("b="+b);"</pre>	
36.	Math.floor(-4.7)	(-5.0)
37.	Math.ceil(3.4) + Math.pow(2,3)	12.0
38.	"ACHIEVEMENT".replace('E', 'A')	ACIAVAMAT
39.	"dedicate".compareTo("devote")	-18
40.	<pre>String s= "4.3756"; int n= s.indexOf("."); int c= Integer.parseInt(s.substring(0,n)); int m= Integer.valueOf(s.substring(n+1)); System.out.println(c); System.out.println(m);"</pre>	4 3756
41.	<pre>String s1= "good"; String s2= "world matters"; String str1= s2.substring(5).replace('t', 'n'); String str2= s1.concat(str1);"</pre>	str1=manners str2= good manners
42.	<pre>String x= "Computer"; String y= "Applications"; i) System.out.println(x.substring(1,5)); ii) System.out.println(x.indexOf(x.charAt(4))); iii) System.out.println(y+ x.substring(5)); iv) System.out.println(x.equals(y));"</pre>	i) ompu ii) 4 iii) Applicationster iv) false
43.	<pre>String s= "Today is Test"; System.out.println(s.indexOf("T")); System.out.println(s.substring(0,7)+ ""+ "Holiday");"</pre>	0 Today i Holiday
44.	<pre>String arr[]= { "delhi", "chennai", "mumbai", "lucknow", "jaipur" System.out.println(arr[0].length>arr[3].length()); System.out.print(arr[4].substring(0,3));"</pre>	false JAI
45.	<pre>String x[]= { "SAMSUNG", "NOKIA", "SONY", "MICROMAX", "BLACKBERRY"} i) System.out.println(x[1]); ii) System.out.println(x[3].length());"</pre>	i)NOKIA ii) 8

46.	String str1= "great" , str2="minds" System.out.println(str1.substring(0,2).concat(str2.substring(1))); System.out.println(("WH"+(str.substring(2).toUpperCase())));	Grinds WHEAT
47.	String s= "1001"; int x= Integer.valueOf(s); double y= Double.valueOf(s); System.out.println("x="+x); System.out.println("y="+y);	x= 1001 y=1001.0
48.	System.out.println("The king said \"Begin at the beginning !\" "to me");	The king said " Begin at the beginning!" to me.
49.	int n[]= {1,2,3,5,7,9,13,16}. What are the values of x and y if: x= Math.pow(n[4],n[2]); y=Math.sqrt(n[5]+n[7]);	x=343.0 y=5.0
50.	String x[]= { "SAMSUNG" , "NOKIA" , "SONY" , "MICROMAX" , "BLACKBERRY"}; System.out.println(x[1]); System.out.println(x[3].length());	NOKIA 8
51.	Name the methods of Scanner class that includes: i) used to input an integer data from standard input stream ii) used to input a String data from the standard input stream	i) nextInt() ii) next()
52.	char a[][]= new char [5][5]; int i,j; for (i=0; i<4 ; i++) { for (j=0, j<4; j++) { if (i==j i+j==3) a[i][j]= 'A'; else a[i][j]= ' '; } for (i=0; i<4 ; i++) { for (j=0;j<4;j++) System.out.print(a[i][j]); System.out.println(); } }	A A AA AA A A
53.	int a[]={10,11,12,13,15}; a[4]= ++a[2]; a[3]=a[2]-a[1]; System.out.print(a[3] + " " + a[4];	2 13
54.	int i,j,a[][]= new int[3][2]; for (i=0; i<3 ; i++) {	3 2 2 4

	<pre> for (j=0; j<3;j++) { a[i][j]= i*j +2; } } System.out.println(a[1][1]+ '\t' + a[0][1] +"\t" + a[2][0] + '\t' + a[2][1] </pre>	
55.	<pre> int a[][]= {{1,2} , {3,4}}; int i,j; for (i=0;i<2;i++) { for (j=0;j<2 ;j++) System.out.println(a[i][j]); } </pre>	1 2 3 4
56.	<pre> public void sampleMethod() { for (int i=0; i<3;i++) { for (int j=0;j<2;j++) { int n= (int)(Math.random()*10); System.out.println(n); }}} </pre> <p>1) How many times the loop execute 2) What is the possible range of values stored in variable n</p>	1) Outer loop will execute 3 times whi inner loop will execute 6 times. 2) 0 to 9
57.	<pre> int i; for(i=5;i>10;i++) System.out.println(i); System.out.println(i*4); </pre>	0 times, since the condition is false Output: 20
58.	<pre> int a=63,b=36; boolean x= (a>b)?true:false; int y= (a<b)?a:b; what will be the values of x and y? </pre>	x=true y=36
59.	<pre> Rewrite using ternary operator if(bill>1000) discount= bill*10.0/100; else discount= bill*5.0/100; </pre>	discount=bill>1000?(bill*10.0/100):(b /100);
60.	<pre> int a,b; for(a=6 ,b=4 ; a<=24 ; a=a+6) { if(a%b==0) break; } </pre>	Loop will execute 2 times.

61.	(imp) int m=2,n=15 ; for(int i=1; i<5; i++); m++; --n; System.out.println("m="+m); System.out.println("n="+n);	m=3; n=14
62.	char x= 'A'; int m; m= (x== 'a')? 'A': 'a'; System.out.println("m="+m);	m=65
63.	When, i) opn= 'b' ii) opn= 'x' iii) opn= 'a' switch(opn) { case 'a'; System.out.println("Platform Independen"); break; case 'b': System.out.println("Object oriented"); case 'c': System.out.println("Robust and Secure"); break; default: System.out.println(" Wrong input"); }	i) Object Oriented Robust and Secure ii) Wrong input iii) Platform independent
64.	Convert the following into switch case: if (var==1) System.out.println("good"); else if (var==2) System.out.println("better"); else if (var==3) System.out.println("best"); else System.out.println("invalid");	switch(var) { case 1: System.out.println("good"); break; case 2: System.out.println("better"); break; case 3: System.out.println("best"); break; default: System.out.println("invalid"); }
65.	x=5; y=50; while(x<=y) { y=y/x; System.out.println(y); }	10 2
66.	for (int i=3; i<=4 ; i++) { for (int j=2; j<i ; j++) { System.out.print("");	i) 3 times ii) WIN WIN

	<pre> } System.out.println("WIN"); i) How many times the inner loop executes? ii) Write the output </pre>	
67.	System.out.println(Math.abs(-999));	999
68.	<pre> class test { void main() { int x,y; if(x==y) System.out.print("1"); else System.out.print("0"); } } </pre>	1
69.	<pre> int x=1; while (x<=10) { if(x==3) break; System.out.print(x); x++; } </pre>	12
70.	<pre> int k=2 , x=1; while (k<=10) { x= x+k; System.out.print(x); k+=2; } </pre>	3 7 13 21 31
71.	<pre> for (int k=1; k<=5 ; k++) { if (k==5) System.out.print(k); else System.out.print(k*k); } </pre>	1 4 9 16 25
72.	<pre> while (ch<= 'z') { System.out.print(ch); ch+=1; } </pre>	a b c d e . . z
73.	<pre> int p=200; while(true) </pre>	The loop will execute 7 times and the output will be 80.

	<pre> { if(p<100) break; p=p-20; } System.out.println(p); How many times the loop will execute and what will be the output </pre>	
74.	<pre> for (int i=11;i<=20;i++) { if (i>13 && i<19) { continue; } System.out.println(i) } </pre>	11 12 13 19 20
75.	<pre> int a=1; while (a<=6) { if(a>3) { break; } System.out.println(a); a++; } </pre>	1 2 3
76.	<pre> public class output1 { public static void main(String args[]) { int a=0; int sum=0; while(a<=7) { sum+=a; if(a>4) { break; } a++; } System.out.println(sum); } } </pre>	15
77.	<pre> int p[]={ 7,3,6, 10, 4 ,6 ,9} int sum=0,a=-1; while(a<6) </pre>	26

	<pre> { a++; if(a==3 a==6) { continue; } sum+=p[a]; } System.out.println(sum); </pre>	
78.	<pre> int a=7,sum=0; do { sum+=a; a++; } while(a<6); System.out.println(sum); </pre>	7
79.	<pre> int i=3,sum=0 do { sum+=i; i++; } while(i<7) System.out.println(sum); </pre>	18
80.	<pre> int i; for (i=1; i<10;i++) System.out.print(i); </pre>	123456789
81.	<pre> int i; for(i=1;;i++) System.out.print(i); </pre>	12345... infinite loop
82.	<pre> int i=5; while(i>1) { System.out.println(i); i--; } </pre>	5 4 3 2
83.	<pre> char a[]={ 'A' , 'I' , 'D' , 'N' , 'I' } int l,i; l=a.length; i=l-1; while (i>=0) [System.out.print(a[i]); i--; </pre>	INDIA

	}	
84.	if (income<=10000) tax=0; else tax=12;	tax=(income<=10000)?0:12;
85.	if (var==10) t=50; else t=120;	t=(var==10)?50:120;
86.	String grade; if (mark>=90) grade= "A"; else if marks>=80) grade= "B"; else grade= "C";	String grade=(marks>=90)? "A":(mark>=80)? "B": "C";
87.	Write a single line ternary statement to find max out of 3 numbers. max= (a>b)?((a>c)?a:c):((b>c)?b:c)	
88.	Rewrite using while loop int f=1, i; for (i=1; i<5 ; i++) { f*i; System.out.println(f); }	int i=1,f=1; while (i<=5) { f*=i; System.out.println(f); i++; }
89.	How many times will the following loop execute? What will be its return value int x=2,y=50; do { ++x; y-=x++; } while (x<=10) return y;	4 times 26
90.	int x,c; for(x=10,c=20;c>=10; c=c-2) x++;	int x=10,c=20; do { x++; c=c-2; } while (c>=10);